

**Amendments to the Claims:** This listing of claims will replace all prior versions, and listings, of claims in the application

Listing of Claims:

1. (Currently Amended) A method of task management comprising the steps of:

- a. receiving one or more tasks to be executed;
- b. atomizing the one or more tasks into one or more atomic sub-tasks; and

c. ~~designating access rights~~assigning protection attributes to indicating a memory range of a shared memory one or more computing resources for each respective atomic sub-task of the one or more atomic sub-tasks such that each respective sub-task is executed by one of a plurality of processors which inherits access rights to the shared memory indicated by the protection attributes corresponding to the respective atomic sub-task.

2. (Original) The method according to claim 1, further comprising the step of:

- d. scheduling the one or more atomic sub-tasks into a central task queue.

3. (Currently Amended) The method according to claim 2, wherein the step of scheduling the one or more atomic sub-tasks into ~~at~~the central task queue is done according to one or both of temporal and priority considerations.

4. (Currently Amended) The method according to claim 2, further comprising the step of:

e. obtaining from a first idle processor of ~~at~~the plurality of processors a first atomic sub-task from the central task queue, the first idle processor thereby inheriting the access rights to a first memory range of the shared memory one or more computing resources ~~of the first atomic sub-task in executing the first atomic sub-task.~~

5. (Currently Amended) The method according to claim 4, further comprising the step of:

f. obtaining from a further idle processor of the plurality of processors a further atomic sub-task from the central task queue, the further idle processor thereby inheriting the access rights to a further memory range of the shared memory~~one or more computing resources of the further atomic sub-task~~ in executing the further atomic sub-task.

6. (Original) The method according to claim 5, wherein steps e and f are repeated until there are no further idle processors or no further atomic sub-tasks in the central task queue.

7. (Original) The method according to claim 6, further comprising the step of combining one or more atomic results of execution of each atomic sub-task corresponding to a task into a result of the task.

8. (Currently Amended) A task-based library for processor management, comprising:

means for receiving tasks to be executed;

a task atomizer for atomizing the tasks into one or more atomic sub-tasks; and

a plurality of processors for executing the one or more atomic sub-tasks; and

an access rights generator for assigning protection attributes indicating a memory range of a shared memory for each respective atomic sub-task of the one or more atomic sub-tasks such that each respective sub-task is executed by one of the plurality of processors which inherits access rights to the shared memory indicated by the protection attributes corresponding to the respective atomic sub-task~~to the tasks designating access rights to one or more processing resources to be used in executing the tasks.~~

9. (Original) The task-based library of claim 8, further comprising a central task queue for storing the one or more atomic sub-tasks waiting to be executed.

10. (Original) The task-based library of claim 9, further comprising a task scheduler for arranging the one or more atomic sub-tasks in the central task queue.

11. (Original) The task-based library of claim 10, further comprising a combiner for combining execution results of the one or more atomic sub-tasks into an execution result of a task.

12. (Canceled)

13. (Original) The task-based library of claim 8, further comprising a combiner for combining execution results of the one or more atomic sub-tasks into an execution result of a task.

14-15. (Canceled)

16. (Currently Amended) A method of task management comprising the steps of:

a. receiving one or more tasks to be executed;

b. atomizing the one or more tasks into one or more atomic sub-tasks;

c. assigning protection attributes indicating a memory range of a shared memory for each respective atomic sub-task of the one or more atomic sub-tasks such that each respective sub-task is executed by an idle processor of a plurality of processors which inherits access rights to the shared memory indicated by the protection attributes corresponding to the respective atomic sub-task~~designating access rights to one or more computing resources for each atomic sub-task of the one or more atomic sub-tasks;~~

d. scheduling the one or more atomic sub-tasks into a central task queue according to one or both of temporal and priority considerations;

e. obtaining via a first idle processor of ~~a~~the plurality of processors a first atomic sub-task from the central task queue for execution of the first atomic sub-task, ~~the first idle processor inheriting the access rights to one or more computing resources of the first atomic sub-task in executing the first atomic sub-task; and~~

f. obtaining via a further idle processor of the plurality of processors a further atomic sub-task from the central task queue for execution of the further atomic sub-task, ~~the further idle processor inheriting the access rights to one or more computing resources of the further atomic sub-task in executing the further atomic sub-task.~~

17. (Original) The method according to claim 16, further comprising the step of repeating steps e and f until there are no further idle processors or no further atomic sub-tasks in the central task queue.

18. (Original) The method according to claim 17, further comprising the step of combining one or more atomic results of execution of each atomic sub-task corresponding to a task into a result of the task.

19. (Currently Amended) The method according to claim 16, wherein the step d of scheduling the one or more atomic sub-tasks into ~~a~~the central task queue is done according to temporal considerations.

20. (Currently Amended) The method according to claim 16, wherein the step d of scheduling the one or more atomic sub-tasks into ~~a~~the central task queue is done according to priority considerations.